

Design of Non-Binary Quasi-Cyclic LDPC Codes by ACE Optimization

Alex Bazarsky, Noam Presman and Simon Litsyn

School of Electrical Engineering, Tel Aviv University, Ramat Aviv 69978 Israel

e-mail: {bazarsky,presmann,litsyn}@eng.tau.ac.il

Abstract—An algorithm for constructing Tanner graphs of non-binary irregular quasi-cyclic LDPC codes is introduced. It employs a new method for selection of edge labels allowing control over the code's non-binary ACE spectrum and resulting in low error-floor. The efficiency of the algorithm is demonstrated by generating good codes of short to moderate length over small fields, outperforming codes generated by the known methods.

I. INTRODUCTION

LDPC codes introduced by Gallager [1] are excellent error correcting codes, which are being used in many modern applications. Non-binary (NB) LDPC codes exhibit better error correcting performance compared to the binary ones [2]. As the field size grows, error correcting capability improves, at the price of increasing decoding complexity.

Binary LDPC codes constructed by quasi-cyclic (QC) lifting of a base-graph have a structure that can be utilized in an efficient implementation of both the encoder and the iterative decoder [3], [4], [5]. NB QC codes based on α -multiplied circulant permutation matrices have similar properties [6].

In LDPC codes, the error-floor is induced by the presence of small combinatorial structures in the Tanner graph (e.g. stopping-sets, trapping-sets, etc.). These structures always contain cycles, therefore manipulating the parameters of the cycles also affects the error-floor. The parameters of importance here are the cycles' length and connectivity, manifested by their approximate cycle extrinsic message degree (ACE). For binary codes, the error-floor can be reduced significantly by removing short cycles having small ACE value from the graph [7], [8], [9], [10]. For QC codes, the ACE-constrained construction becomes more computationally efficient by utilizing the relation between cycles in the protograph and their realization in the lifted graph. Based on this idea, Asvadi *et al.* [11] introduced an algorithm for design of irregular binary QC codes with an excellent error-correcting performance.

NB LDPC codes are conventionally constructed by first obtaining a binary mother parity check-matrix H_b , and then replacing the non-zero elements of H_b by non-zero values from the field, often referred to as labels. The label assignment is performed either randomly or intelligently (by meeting some design criteria).

Poulliat *et al.* [8] designed regular NB $(2, d_c)$ codes (cycle codes) by first using the progressive edge growth (PEG) algorithm [12] to construct H_b having an associated Tanner graph with large girth. Then, they introduced a method for label

assignment based on cycle cancelation, resulting in low error-floor codes. Peng and Chen extended this idea for the design of NB QC regular cycle codes [13]. Other design algorithms for NB QC codes were also explored recently [14] [15].

In this paper, we use a relation between the protograph and the QC lifted graph, to produce good irregular NB QC LDPC codes. Our design involves a new method to select edge labels which constrain the code's NB ACE spectrum, resulting in improved performance. We demonstrate the efficiency of this algorithm by constructing good codes of short to moderate length over small fields. Such codes are practical due to their moderate decoding complexity. Note, that irregular profiles achieve better error correcting performance compared to regular ones for codes over small fields [2] [12], which motivates our ACE based design.

The paper is organized as follows. In Section II, we begin by presenting the relevant background and notations that are used throughout. In Section III, we present our NB QC code construction method. The performance of codes generated by the method is demonstrated by simulations in Section IV.

II. PRELIMINARIES

Throughout we use the following notations. For an integer $n > 0$, let $[n] = \{1, 2, \dots, n\}$. For two integers a, b , the remainder of the division of a by b is denoted by $R_b[a]$. For two vectors $\underline{u}, \underline{v}$ of length ℓ , $\underline{u} \geq \underline{v}$ iff $u_i \geq v_i, \forall i \in [\ell]$.

A. LDPC Codes

A binary LDPC code of length n is a linear block code defined by a binary parity-check matrix $H_{m \times n}$. The code can be equivalently represented by a bipartite Tanner graph $G = (V \cup C, E)$, where the set V consists of variable nodes $v_i, i \in [n]$, and the set C consists of check nodes $c_j, j \in [m]$. An edge connects a variable node v_i to a check node c_j iff $H_{j,i} = 1$. The degree distribution of the code is represented by two polynomials: $\Lambda(x) = \sum_{i=2}^{d_v} \Lambda_i x^i$ for the variable nodes and $\Gamma(x) = \sum_{i=2}^{d_c} \Gamma_i x^i$ for the check nodes, where d_v (d_c) is the maximum variable (check) node degree, and where Λ_i (Γ_i) is the number of degree i variable (check) nodes. If $\Lambda(x) = n \cdot x^{d_v}$ and $\Gamma(x) = m \cdot x^{d_c}$, the code is called regular (d_v, d_c) LDPC. Otherwise, the code is called irregular LDPC.

An NB LDPC code over $GF(q)$ is defined by a parity-check matrix H with elements from the field ($q = 2^r, r > 1$). The Tanner graph of the code has labels on its edges which are the corresponding non-zero entries from H . For such a code, we

define a binary matrix H_b of the same dimensions as H , such that each entry in H_b is 1 iff the corresponding entry in H is non-zero. H_b is referred to as the *binary mother matrix* of the code.

B. QC Lifted Codes

Definition 1 (Lifted graph): Let $\hat{G} = (\hat{V} \cup \hat{C}, \hat{E})$ be a Tanner graph. For each $(v, c) \in \hat{E}$ define a permutation $\pi_{(v,c)}$ on the set $[Z]$. To each $v \in \hat{V}$ ($c \in \hat{C}$) we generate a set of Z duplicates v_i (c_i), $i \in [Z]$. Then $G = (V \cup C, E)$ is a Z -lifted graph of \hat{G} , associated with these permutations, if $V = \{v_i | v \in \hat{V}, i \in [Z]\}$, $C = \{c_i | c \in \hat{C}, i \in [Z]\}$ and $E = \{(v_i, c_j) | (v, c) \in \hat{E} \wedge \pi_{(v,c)}(i) = j\}$.

The graph \hat{G} is called a base-graph or a protograph. When all the edge permutations in G are cyclic shifts of $[Z]$, then G is called Z -lifted QC graph. In this case, a binary LDPC code associated with G , has a compact block-representation of its parity check matrix H , based on the parity check matrix \hat{H} associated with the protograph. In this representation, each entry is replaced by a $Z \times Z$ matrix as follows. The zero entries are replaced by zero matrices. Each non-zero entry is replaced by a (Z, d) -circulant permutation matrix (CPM), defined below, if a right circular shift of d places is assigned to it.

Definition 2 (CPM): A (Z, d) -CPM is formed by a circular shift to the right by d places of the columns of the $Z \times Z$ identity matrix.

In an NB QC Z -lifted code, each non-zero element of the protograph parity check matrix, corresponding to an edge e , is replaced by a $(Z, \lambda_e, \rho_e, d_e)$ -multiplied CPM (MCPM), as defined below [6], [13].

Definition 3 (MCPM): Let $(q-1)|\lambda Z$ and α is a primitive element of $GF(q)$. A (Z, λ, ρ, d) -MCPM over $GF(q)$, is a $Z \times Z$ matrix, with underlying binary mother matrix (Z, d) -CPM. Furthermore, for each row i of the MCPM $i \in [Z]$, the single non-zero element is $\alpha^{\rho+(i-1)\cdot\lambda}$.

Note, that in a CPM, each row is an α^λ -multiplied circular right shift of the row above it. This is also true for the first row, where the row "above it" is defined to be the last row.

C. Cycles and ACE

Cycles in the Tanner graph are known to influence the error-floor of iterative decoders (e.g. stopping-sets, trapping-sets, etc.). Important combinatorial characteristics of a cycle are its length and its extrinsic message degree (EMD), which is the number of check-nodes that are connected to the variables of the cycle by only one edge. In this paper, we use for simplicity, the approximate cycle EMD (ACE), defined as $\sum_{v_i} (d_{v_i} - 2)$, where d_v is the degree of a node v , and the summation is over all the variable-nodes of the cycle. A code with long cycles and large ACE usually exhibits smaller error-floor compared to a code with shorter cycles or smaller ACE. This notion motivates the following definition.

Definition 4 (ACE spectrum): For an LDPC code represented by a Tanner graph G , the ℓ -depth ACE spectrum is $\bar{\tau}^{(b)}(G) = (\tau_2^{(b)}, \tau_4^{(b)}, \dots, \tau_\ell^{(b)})$ where $\tau_i^{(b)}$, $i \leq \ell$, is the

minimum ACE value of any cycle of length i in G . G achieves an ℓ -depth ACE constraint $\hat{\tau}^{(b)} = (\hat{\tau}_2^{(b)}, \hat{\tau}_4^{(b)}, \dots, \hat{\tau}_\ell^{(b)})$ if $\bar{\tau}^{(b)}(G) \geq \hat{\tau}^{(b)}$.

To lower error-floor, it is beneficial to achieve higher ACE spectrum values for cycles of lower length.

We now discuss the relationship between cycles in the protograph \hat{G} and the resultant QC Z -lifted graph G . Let \mathcal{C} be a cycle in \hat{G} of even length ℓ , being a sequence of edges $\{e_i\}_{i=1}^\ell$, having ACE τ . Assume that in G , for each edge e_i we used a (Z, d_i) -CPM, $i \in [\ell]$. The order of \mathcal{C} in G is defined as $O(\mathcal{C}) = Z / \gcd(Z, d)$, where $d = R_Z \left[\sum_{i=0}^{\ell-1} (-1)^i d_{i+1} \right]$ is called the *total shift* of the cycle \mathcal{C} . It is easy to see that the lifted nodes and edges of \mathcal{C} in G form a union of $\gcd(Z, d)$ cycles, each one of them having length $\ell \cdot O(\mathcal{C})$ and ACE $\tau \cdot O(\mathcal{C})$. Moreover, every cycle in G corresponds to a cycle in \hat{G} . Therefore, the ACE spectrum of G can be easily derived from the knowledge about cycles in \hat{G} and the shifts of its edges.

In NB LDPC codes each cycle also has a meaningful algebraic structure, defined by the labels on its edges. A simple and minimal cycle (i.e. that does not contain a cycle being a subset of its nodes), \mathcal{C} , of length ℓ in an NB Tanner graph G , with parity check matrix H can be represented by an $\ell/2 \times \ell/2$ matrix denoted B . Here, B is the sub-matrix of H with rows and columns that correspond to the check nodes and variable nodes that are in \mathcal{C} . Without loss of generality, we can assume that B has the following canonical form. The i th row, $i \in [\ell/2 - 1]$ is of the form $[\mathbf{0}_{i-1}, \beta_{2(i-1)}, \beta_{2(i-1)+1}, \mathbf{0}_{\ell/2-i-1}]$ and the last row is $[\beta_{\ell-1}, \mathbf{0}_{\ell/2-2}, \beta_{\ell-2}]$. Here, for $i \geq 0$, $\mathbf{0}_i$ is the zero-vector of length i (in case $i = 0$, this is the empty vector) and $\{\beta_i\}_{i=0}^{\ell-1}$ are non-zero elements of $GF(q)$.

In LDPC cycle codes, studied by Poulliat *et al.* [8], the variable nodes of every simple and minimal cycle form support of a codeword, unless its corresponding matrix B is full-rank. In a canonical form of B , this full-rank condition (FRC) is equivalent to the following

$$(FRC) : \prod_{i=0}^{\ell/2-1} \beta_{2i+1} \neq \prod_{i=0}^{\ell/2-1} \beta_{2i}. \quad (1)$$

Therefore, to avoid low-weight codewords in cycle codes (causing high error-floor), Poulliat *et al.*, assign the labels of the NB code, so that cycles of short length fulfill the FRC. A cycle \mathcal{C} , that satisfies the FRC is said to be "canceled".

We argue that even for general NB irregular codes, assigning labels such that a cycle \mathcal{C} is canceled, should reduce the probability that the BP iterative decoder fails to converge due to errors in the variables of the cycle. The intuition behind it is that the constraints of the matrix B corresponding to \mathcal{C} , imply a local-code on the variables of \mathcal{C} with a single codeword (the zero-codeword) iff \mathcal{C} is canceled. Since the iterative decoder is local in its behavior, if \mathcal{C} is not canceled, the decoder could be misled to converge to one of the wrong codewords of the local-code of \mathcal{C} . The cycle's extrinsic check nodes may prevent such an erroneous convergence. Having more such extrinsic

check nodes should increase the chance to overcome errors in the variables of \mathcal{C} . Hence, because label assignment can cancel only a limited number of cycles, it seems reasonable to prefer canceling the shorter ones with low ACE. This notion justifies the next useful definition.

Definition 5 (NB ACE spectrum): For an NB LDPC code, represented by a Tanner graph G , the ℓ -depth NB ACE spectrum is $\bar{\tau}^{(nb)}(G) = (\tau_2^{(nb)}, \tau_4^{(nb)}, \dots, \tau_\ell^{(nb)})$ where $\tau_i^{(nb)}$, $i \leq \ell$, is the minimum ACE value of any non-canceled cycle of length i in G . G achieves an ℓ -depth NB ACE spectrum $\hat{\tau}^{(nb)} = \{\hat{\tau}_2^{(nb)}, \hat{\tau}_4^{(nb)}, \dots, \hat{\tau}_\ell^{(nb)}\}$ if $\bar{\tau}^{(nb)}(G) \geq \hat{\tau}^{(nb)}$.

Peng and Chen [13] showed that for NB QC Z -lifted codes, the FRC of the lifted-cycles resulting from a protograph cycle \mathcal{C} can be simply expressed through the parameters of the MCPMs assigned to the edges of \mathcal{C} .

Theorem 1 ([13]): Suppose a simple and minimal cycle \mathcal{C} of length ℓ on the protograph is represented by a matrix B in its canonical form. Let \tilde{B} be the QC Z -lifted representation of B

$$\tilde{B} = \begin{pmatrix} P_0 & P_1 & 0 & \dots & 0 & 0 \\ 0 & P_2 & P_3 & \dots & 0 & 0 \\ 0 & 0 & P_4 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ddots & P_{\ell-4} & P_{\ell-3} \\ P_{\ell-1} & 0 & 0 & \dots & 0 & P_{\ell-2} \end{pmatrix}, \quad (2)$$

where P_i is $(Z, \lambda, \rho_i, d_i)$ -MCPM. The FRC condition for the cycles induced by \mathcal{C} in the lifted graph (each one of length $O(\mathcal{C}) \cdot \ell$) is

$$O(\mathcal{C}) \cdot \sum_{i=0}^{\ell-1} (-1)^i \rho_i \neq 0 \pmod{q-1}. \quad (3)$$

Note, that Peng and Chen required that $(q-1) = Z \cdot \lambda$, however their proof is still valid even in a more general case of $(q-1) | Z \cdot \lambda$. The combinatorial and algebraic connections between the cycles of the protograph and the cycles of the lifted-graph, are a key to the efficient algorithms we present in the next section.

III. NON-BINARY QC ACE CONSTRAINED CODE CONSTRUCTION

In this section, we introduce a code construction algorithm. The inputs to the algorithm are the degree profile of a protograph \hat{G} , a lifting order Z and a field size q . The algorithm is also given two ℓ -depth ACE spectrum constraints, $\hat{\tau}^{(b)} = (\hat{\tau}_2^{(b)}, \dots, \hat{\tau}_\ell^{(b)})$ and $\hat{\tau}^{(nb)} = (\hat{\tau}_2^{(nb)}, \dots, \hat{\tau}_\ell^{(nb)})$, such that $\hat{\tau}^{(nb)} \geq \hat{\tau}^{(b)}$. The output is G , a QC Z -lifting of \hat{G} with labels from $GF(q) \setminus \{0\}$, which is NB ACE constrained by $\hat{\tau}^{(nb)}$ and its binary mother matrix is ACE constrained by $\hat{\tau}^{(b)}$, if both spectrums are achievable. The algorithm consists of the following steps:

Step 1: Construct a good protograph \hat{G} by any protograph selection method (e.g [9]).

Step 2: Construct a QC Z -lifted graph of \hat{G} , that is $\hat{\tau}^{(b)}$ -ACE constrained, by carefully choosing for each edge of \hat{G} the cyclic shift of its Z copies (see Subsection III-A). This graph is the binary mother matrix of the output G .

Step 3: Assign labels to the edges of the mother matrix, such that the resultant NB labeled graph G is NB ACE constrained by $\hat{\tau}^{(nb)}$. This label assignment ensures that all the cycles in G that violate $\hat{\tau}^{(nb)}$ satisfy the FRC (see Subsection III-B).

Good achievable constraint vectors $\hat{\tau}^{(b)}$, $\hat{\tau}^{(nb)}$ and their depth ℓ may be found by the following heuristic search. Find initial constraints $\hat{\tau}^{(b)}$, $\hat{\tau}^{(nb)}$ by first running the above algorithm with no constraints, and retrieve the spectra of the resultant graph G . Then, attempt to improve the spectra by increasing their depth ℓ or increasing their components and rerun the algorithm with the amended spectrum. Repeat this procedure (amending the spectra and rerunning the algorithm) until no further improvement is achieved (i.e. the algorithm fails to find a graph that achieves the constraints). Note, that since it is not always possible to determine which spectrum is better (see e.g. [9, Section IV]), the designer is advised in these cases, to generate graphs for each of these competing spectra and choose the best one by a simulation. Furthermore, because of the random nature of the algorithm, it is recommended to run the algorithm several times for each set of good parameters, thereby generating different instances of G satisfying the requirements. Here, again, the best instance, may be chosen by a simulation.

A. Construction of the Binary Mother Matrix

We now describe an algorithm that finds a QC binary code that satisfies certain ACE spectrum constraints. The inputs to the algorithm are a protograph \hat{G} , a lifting factor Z and an ACE spectrum constraint vector $\hat{\tau}^{(b)} = (\hat{\tau}_2^{(b)}, \hat{\tau}_4^{(b)}, \dots, \hat{\tau}_\ell^{(b)})$. The algorithm searches for a QC Z -lifted code with Tanner graph G which achieves $\hat{\tau}^{(b)}$. G is defined by assigning a cyclic shift $d_e \in [Z]$ to each edge e of \hat{G} .

We begin by a preliminary step in which we find all the problematic cycles of \hat{G} which violate $\hat{\tau}^{(b)}$. Denote this set of problematic cycles by S . The lifted versions of the other cycles of \hat{G} will satisfy the $\hat{\tau}^{(b)}$ constraint for any choice of shifts. Next, for each edge e of \hat{G} we enumerate the problematic cycles which include e . We arbitrarily choose initial assignments of d_e for each edge e of \hat{G} (it is recommended to draw these assignments uniformly at random).

The generation of G is iterative. In each iteration, we scan all the edges of \hat{G} in an arbitrary order. For each edge e , we choose a shift $d_e \in [Z]$ that minimizes the number of cycles in S which still violate the $\hat{\tau}^{(b)}$ constraint. If by the end of the iteration, all the lifted versions of the cycles in S satisfy $\hat{\tau}^{(b)}$, the algorithm outputs G , otherwise, another iteration may be initiated. Note, that there is no guarantee that the algorithm finds G that achieves $\hat{\tau}^{(b)}$ even if such G exists. If after a predefined number of iterations G is not found, the designer may consider choosing a different initial assignment of the shifts d_e or changing the order in which the edges are visited, and repeat the iterative part of the algorithm. Our experience shows that usually when G exists, it is found after a small number of iterations.

Our method differs from the algorithm of Asvadi *et al.* [11],

[16] in the following aspects. In [16, Algorithm 1], the cycles in S are sequentially scanned and for each cycle, the shifts d_e of the “unshifted edges” are assigned whereas in our method, the objects being treated are the edges. Furthermore, we allow reassignment of d_e in case the ACE spectrum constraint was not satisfied after the first iteration. Our experience indicates that in many cases our proposed algorithm finds G faster than [16, Algorithm 1]. Peng and Chen [13], suggested to find the binary lifted graph G of cycle codes, having girth ℓ . Note, that this is equivalent to having ACE spectrum constraints $\hat{\tau}^{(b)}$ such that $\hat{\tau}_i^{(b)} = \infty$ for all $i \leq \ell$. Such a spectrum can be achieved only for relatively small ℓ . As a result, longer cycles with low ACE are ignored. Our results indicate that these cycles have an impact on the code’s performance in the error-floor region.

B. Non-Binary ACE Constrained Label Assignment

We now describe the label assignment algorithm to the entries of the binary mother matrix. The inputs to the algorithm are the QC Z -lifted graph G (associated with the binary mother matrix) expressed as the underlying protograph \hat{G} and the selected shifts of its edges. Additional inputs are the NB ACE constraint vector $\hat{\tau}^{(nb)}$ and the field size q . The output of the algorithm is an NB QC code over $GF(q)$ that satisfies $\hat{\tau}^{(nb)}$.

The structure of the algorithm is very similar to the one presented in the previous subsection, therefore we only highlight here the differences between them. We begin by enumerating the set S of problematic cycles in \hat{G} . This time a cycle is problematic if its Z -lift in G violates $\hat{\tau}^{(nb)}$. For each edge e in \hat{G} we choose an initial label ρ_e (preferably at random). We then run the iterative part of the algorithm from Subsection III-A in which we assign labels ρ_e (instead of shifts). Note that, in this case, a problematic cycle violates $\hat{\tau}^{(nb)}$ if its labels do not satisfy (3).

The algorithm we described can be seen as a generalization of the method suggested by Peng and Chen [13] for construction of NB QC lifted graphs of cycle codes. In their algorithm, all cycles up to length ℓ are canceled, which is equivalent to using our algorithm with $\hat{\tau}^{(nb)}$ such that $\hat{\tau}_i^{(nb)} = \infty$ for all $i \leq \ell$. Furthermore, we allow a more flexible choice of the lifting order Z and the CPM parameter λ , by only requiring that $(q-1)|Z\lambda$ (Peng and Chen’s requirement is that $Z|(q-1)$ and $\lambda = (q-1)/Z$).

IV. SIMULATION RESULTS

In this section, we compare the error correcting performance of codes generated by the following techniques: **PEG** - PEG generated mother binary matrix of an irregular code with randomly selected labels [12]. **REG** - PEG generated mother binary matrix of a regular cycle code ($d_v = 2, d_c = 4$) with selective choice of NB labels using the FRC [8]. **GIRTH** - the QC construction presented by Peng and Chen [13] modified to produce irregular codes, where only the girth requirements are taken into account. **ACENB** - our algorithm from Section III. **ACEB** - our algorithm **ACENB** in which Step 3 is replaced by random edge label assignment.

All the codes are simulated over a memoryless binary input AWGN channel using the BPSK modulation and decoded by the iterative NB belief-propagation (BP) algorithm. The maximum number of BP flooding iterations is fixed to 80. All the generated codes are of rate 1/2, and their ensemble properties are summarized in Table I. The degree distributions are selected according to Hu *et al.* [12]. Note, that $\Lambda(x)$ and $\Gamma(x)$ are degrees profiles of the protograph and Z is the lifting order, resulting in an ensemble of QC codes of the specified length in bits. For each ensemble, we use a single protograph matrix (Step 1 in Section III) throughout the various generation methods.

In Table II, the achieved ACE spectra are summarized. For the codes, generated by **ACENB** and **GIRTH**, two spectra are provided. The first one is the spectrum achieved by the binary mother matrix (Step 2 in our code construction). The second one is the NB ACE spectrum achieved by the edge labels assignment (Step 3 in our code construction). For the codes generated by **ACEB**, only one spectrum is provided, since the random label assignment does not take into account any NB ACE spectrum requirement. The achieved ACE spectrum of the codes generated by **GIRTH**, depends only on the codes’ girth, while the NB ACE spectrum depends only on the canceled cycles’ length. Note that the achievable ACE spectrum values grow with the code length.

#	Field	Length [bits]	$\Lambda(x)$	$\Gamma(x)$	Z
1	GF(16)	504	$10x^2 + 2x^3 + 2x^4$	$x^4 + 6x^5$	9
2	GF(8)	1008	$10x^2 + 3x^3 + 3x^4$	$7x^5 + x^6$	16
3	GF(16)	1008	$10x^2 + 2x^3 + 2x^4$	$x^4 + 6x^5$	18
4	GF(16)	1512	$10x^2 + 2x^3 + 2x^4$	$x^4 + 6x^5$	27

TABLE I
ENSEMBLE PROPERTIES OF THE GENERATED CODES

#	GIRTH	ACEB	ACENB
1	$\tau_i = \infty, i \leq 6$ $\tau_i = \infty, i \leq 8$	$(\infty, \infty, \infty, 4)$	$(\infty, \infty, \infty, 4)$ $(\infty, \infty, \infty, \infty, \infty, 4)$
2	$\tau_i = \infty, i \leq 8$ $\tau_i = \infty, i \leq 10$	$(\infty, \infty, \infty, 5, 2)$	$(\infty, \infty, \infty, 6, 2)$ $(\infty, \infty, \infty, \infty, 6, 2)$
3	$\tau_i = \infty, i \leq 8$ $\tau_i = \infty, i \leq 10$	$(\infty, \infty, \infty, \infty, 3, 1)$	$(\infty, \infty, \infty, \infty, 3, 1)$ $(\infty, \infty, \infty, \infty, \infty, 6, 2)$
4	$\tau_i = \infty, i \leq 8$ $\tau_i = \infty, i \leq 10$	$(\infty, \infty, \infty, \infty, 4, 2)$	$(\infty, \infty, \infty, \infty, 4, 2)$ $(\infty, \infty, \infty, \infty, \infty, 9, 3)$

TABLE II
ACE SPECTRA ACHIEVED BY EACH OF THE GENERATED CODES. FOR COLUMNS **ACEB** AND **ACENB**, THE SPECTRUM’S FORMAT IS $(\tau_2, \tau_4, \tau_6, \dots)$.

In Figure 1, the block error rate (BLER) curves of codes from ensembles #2 and #3 from Table I are depicted. As expected, the regular code generated by **REG** is inferior to the other irregular codes. Furthermore, even though the code generated by **PEG** is not constrained by the QC requirement, its performance in the high SNR region is worse than our QC codes. For each ensemble, it’s evident that the code generated by **ACENB**

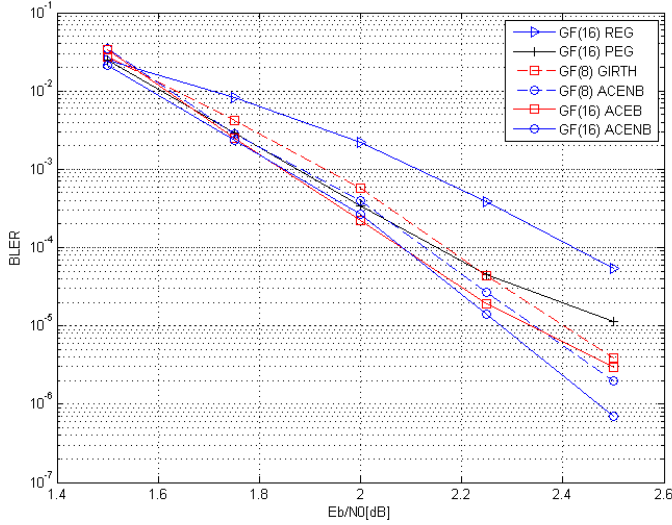


Fig. 1. Simulation of codes having length $n_b = 1008$ and rate $1/2$, constructed by various methods over GF(8) and GF(16)

outperforms the codes generated by the other methods in the high SNR region. Since the codes generated by **ACEB** and **ACENB** have the same binary mother matrix, the error curves highlight the advantage of applying Step 3 in **ACENB** instead of random label assignment. Also, with the increase of the field size, we see an improvement in the high SNR region.

In Figure 2, the BLER curves of codes from ensembles #1 and #4 from Table I are depicted. We also give as a reference the BLER curve of **L20R32A**, a QC code recently generated by Chang *et al.* [14, Figure 10]. Note that, the performance of the code generated by **REG** agrees with the results of Poulliat *et al.* [8, Figure 4], which is simulated using 1000 as the maximum number of BP iterations. To have a fair comparison with the latter code, we use the same limitation on the number of iterations for the codes from ensemble #4. The curves indicate that codes generated by **ACENB** outperform the other codes with similar lengths in the high SNR region. Furthermore, the comparison of the curves corresponding to **ACENB** and **GIRTH** indicates the advantage of ACE driven construction of NB codes.

V. SUMMARY AND CONCLUSIONS

We presented an algorithm to design the Tanner graph of NB QC LDPC codes using ACE constraints for both the generation of the binary mother matrix and the selection of NB labels. Our simulation results indicate that codes generated by this method outperform codes generated by known methods, for different small field sizes and code lengths.

Our method is composed of three separate steps. It is an open question whether combining could be beneficial. This is a matter for future research.

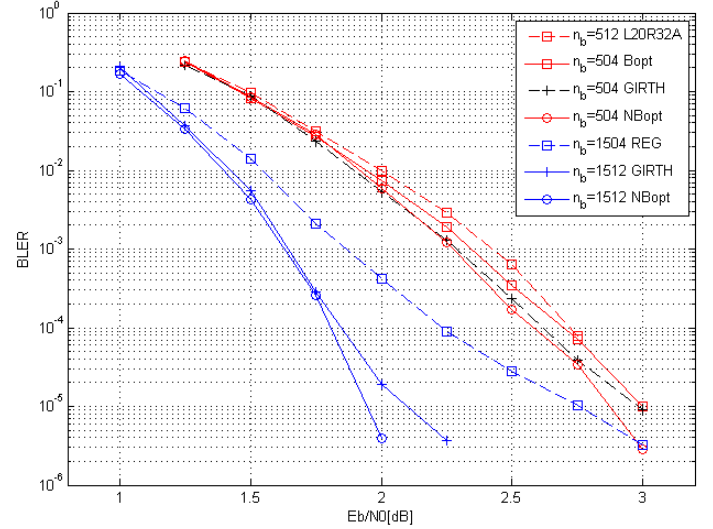


Fig. 2. Simulation of rate $1/2$ codes having lengths around $n_b = 512$ and $n_b = 1504$ over GF(16)

REFERENCES

- [1] R. Gallager, "Low-density parity-check codes," *Information Theory, IRE Transactions on*, vol. 8, no. 1, pp. 21–28, january 1962.
- [2] M. Davey and D. MacKay, "Low-density parity check codes over GF(q)," *Communications Letters, IEEE*, vol. 2, no. 6, pp. 165–167, 1998.
- [3] M. Mansour and N. Shanbhag, "High-throughput LDPC decoders," *IEEE Trans. on Very Large Scale Integration Systems*, vol. 11, no. 6, pp. 976–996, 2003.
- [4] V. Novichkov, H. Jin, and T. Richardson, "Programmable vector processor for irregular LDPC codes," in *38th Annual Conf. on Info. Sciences and Systems*, March 2004.
- [5] J. Thorpe, "Low-density parity-check (ldpc) codes constructed from protographs," *IPN Progress Report, Tech. Rep. 42-154*, 2003.
- [6] L. Zeng, L. Lan, Y. Tai, S. Song, S. Lin, and K. Abdel-Ghaffar, "Constructions of nonbinary quasi-cyclic LDPC codes: A finite field approach," *IEEE Trans. Commun.*, vol. 56, no. 4, pp. 545–554, april 2008.
- [7] T. Tian, C. Jones, J. Villaseñor, and R. Wesel, "Selective avoidance of cycles in irregular LDPC code construction," *IEEE Trans. Commun.*, vol. 52, no. 8, pp. 1242–1247, aug. 2004.
- [8] C. Poulliat, M. Fossorier, and D. Declercq, "Design of regular $(2, d_c)$ -LDPC codes over GF(q) using their binary images," *IEEE Trans. Commun.*, vol. 56, no. 10, pp. 1626–1635, october 2008.
- [9] D. Vukobratovic and V. Senk, "Generalized ACE constrained progressive edge-growth LDPC code design," *IEEE Commun. Lett.*, vol. 12, no. 1, pp. 32–34, january 2008.
- [10] E. Sharon and S. Litsyn, "Constructing LDPC codes by error minimization progressive edge growth," *IEEE Trans. Commun.*, vol. 56, no. 3, pp. 359–368, 2008.
- [11] R. Asvadi, A. Banihashemi, and M. Ahmadian-Attari, "Design of finite-length irregular protograph codes with low error floors over the binary-input AWGN channel using cyclic liftings," *IEEE Trans. Commun.*, vol. 60, no. 4, pp. 902–907, april 2012.
- [12] X.-Y. Hu, E. Eleftheriou, and D. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, jan. 2005.
- [13] R.-H. Peng and R.-R. Chen, "Design of nonbinary quasi-cyclic LDPC cycle codes," in *Information Theory Workshop, 2007. ITW '07. IEEE*, sept. 2007, pp. 13–18.
- [14] B.-Y. Chang, D. Divsalar, and L. Dolecek, "Non-binary protograph-based LDPC codes for short block-lengths," in *Information Theory Workshop (ITW), 2012 IEEE*, Sept., pp. 282–286.
- [15] J. Huang, L. Liu, W. Zhou, and S. Zhou, "Large-girth nonbinary QC-

LDPC codes of various lengths," *IEEE Trans. Commun.*, vol. 58, no. 12, pp. 3436–3447, Dec. 2010.

- [16] R. Asvadi, A. Banihashemi, and M. Ahmadian-Attari, "Lowering the error floor of LDPC codes using cyclic liftings," *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 2213–2224, april 2011.